# Hands-On:
# Add Syncing to Your Application

## Session 122

These are confidential sessions—please refrain from filming or taking pictures.

**Application Technologies**

# Hands-On:
# Add Syncing to Your Application

**Ken Kocienda**
Apple Computer

http://developer.apple.com/wwdc2005/

# Sync makes your application better

# At Home

Favorite Color:

**Desktop computer**

# At Home

Favorite Color:

**Desktop computer**

# At Home

Favorite Color:

**Desktop computer**

# At Work

**Desktop computer**

## At Home

Favorite Color:

**Desktop computer**

## At Work

Favorite Color:

**Desktop computer**

## At Home

Favorite Color:

**Desktop computer**

## At Work

Favorite Color:

**Desktop computer**

# At Home

Favorite Color:

**Desktop computer**

# At Work

Favorite Color:

**Desktop computer**

# SyncServices.framework

# Sync Engine

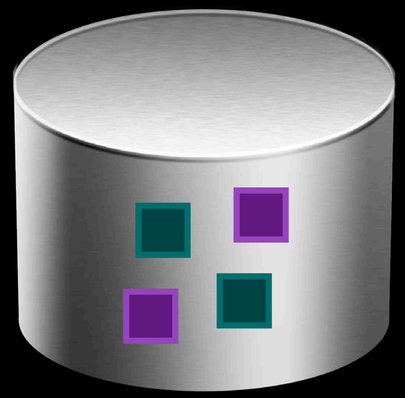
# SyncServices.framework

## Public API for Tiger

# SyncServices.framework

## Here’s how it works

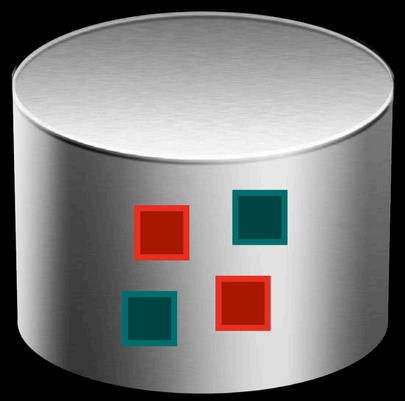
# Sync Engine
## Talks to The Truth



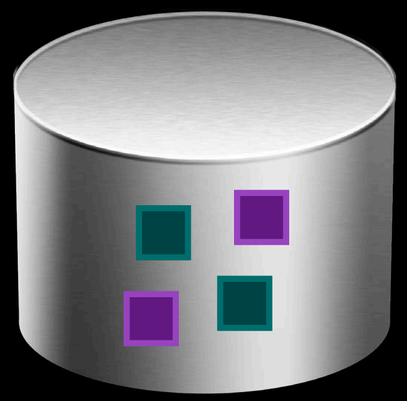
# The Truth
## Stores sync data
## Local / per-user

Application

Sync Engine



Application Data

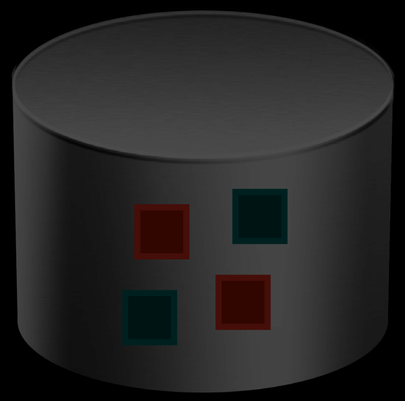


The Truth

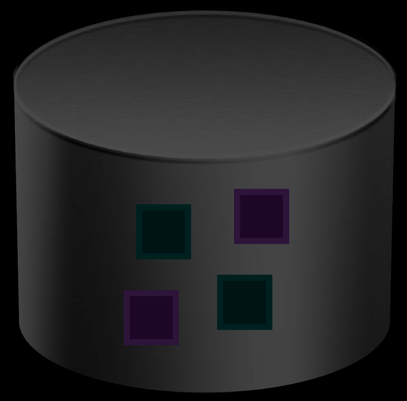
Application

Sync Engine

**Sync application data with The Truth**



Application Data

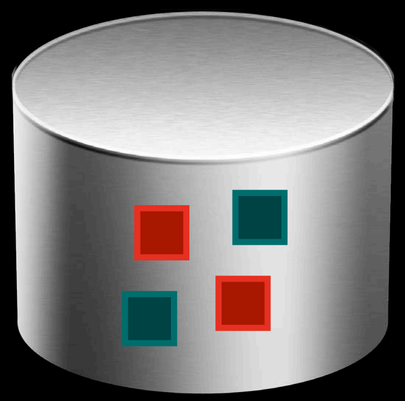


The Truth

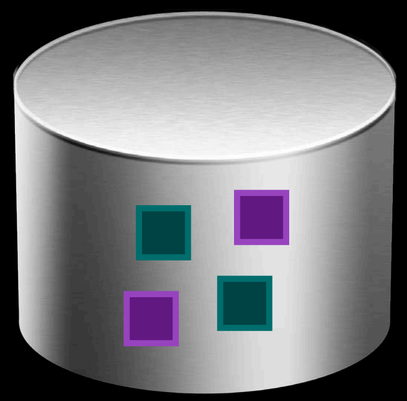
Application

SyncServices API

Sync Engine



Application Data

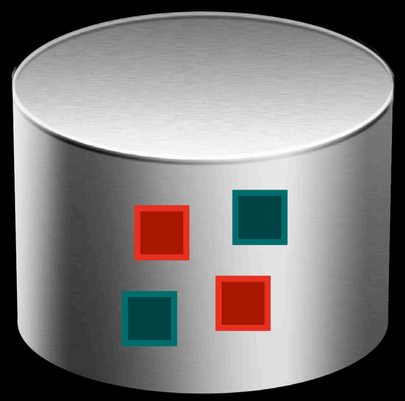


The Truth

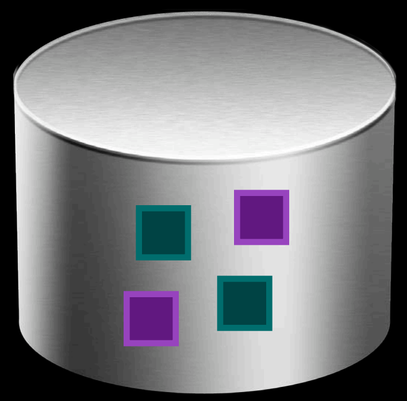
Application

Sync Client

Sync Engine



Application Data

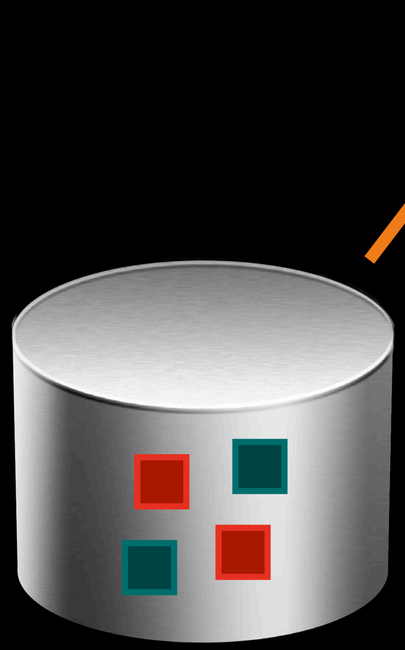


The Truth

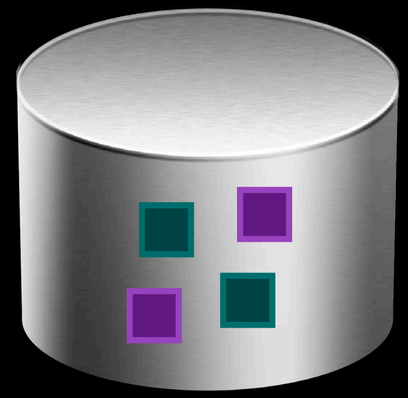
Application

Sync Engine

Sync Client



Application Data

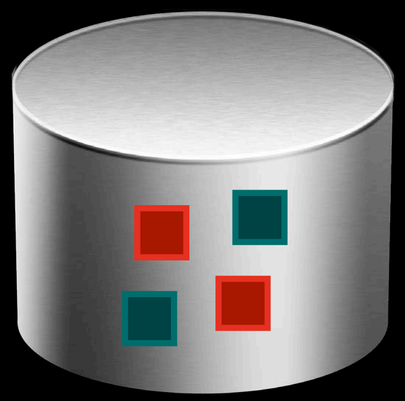


The Truth

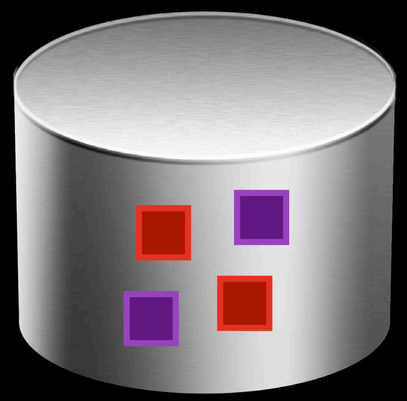
Application

Sync Client

Sync Engine



Application Data

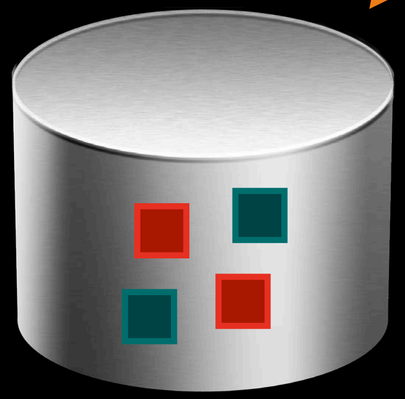


The Truth

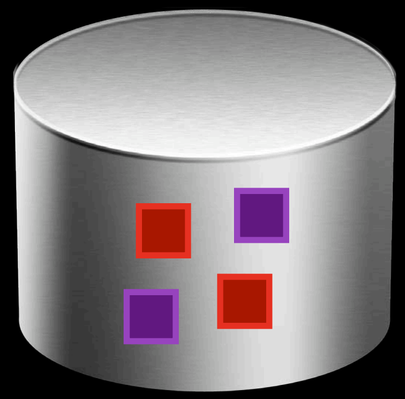
Application

Sync Engine

Sync Client



Application Data

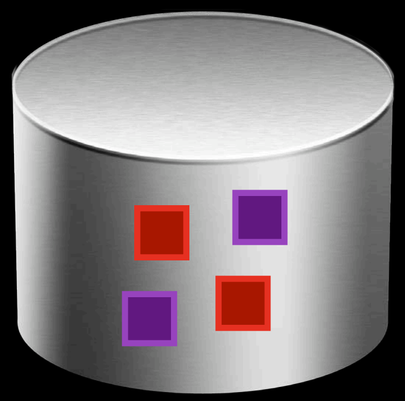


The Truth

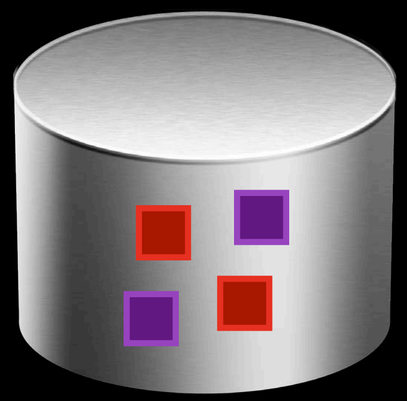
Application

Sync Client

Sync Engine



Application Data



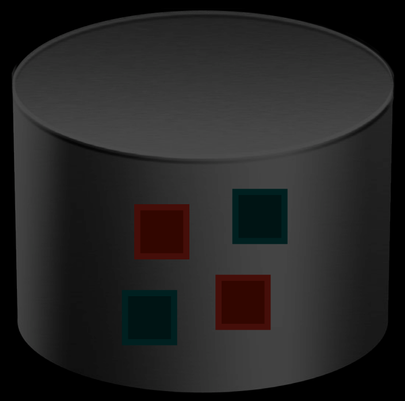
The Truth

Application

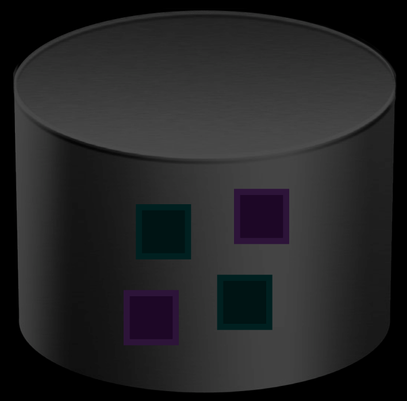
Sync Client

Sync Engine

**Application data in sync with The Truth**



Application Data



The Truth

Application

Sync Client

Sync Engine

## Application data in sync with The Truth

Free!

Conflict management

.Mac sync

Merge data with clients and devices

Data change alert

Application Data

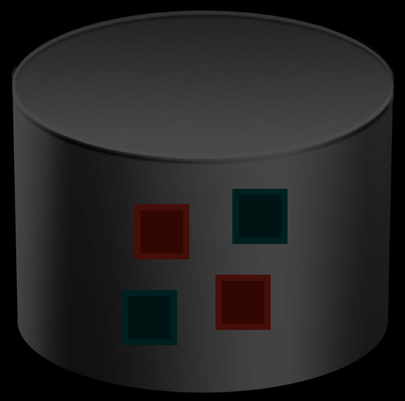
The Truth

Application

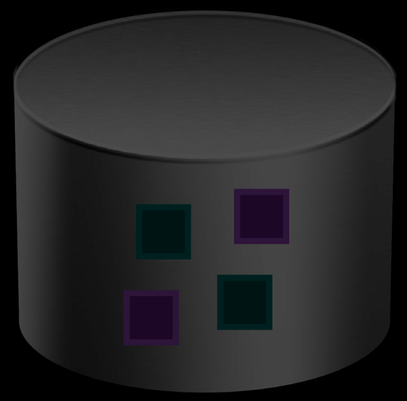
Sync Client

Sync Engine

**Pretty interesting**

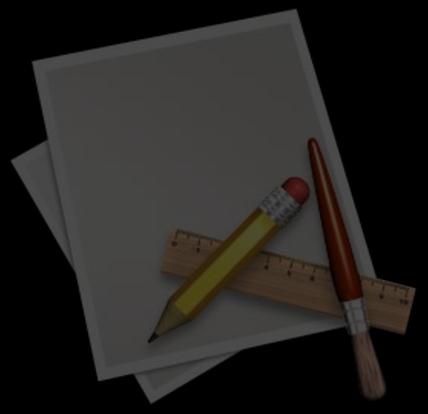


Application Data



The Truth

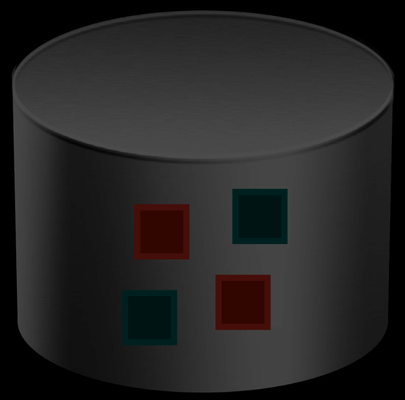
Application



Sync Client

Sync Engine

**But wait, there's more…**



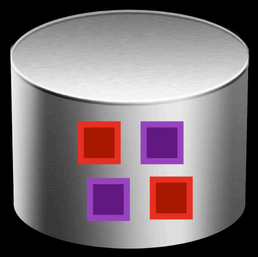
Application Data

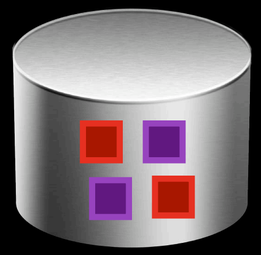
The Truth

iSync

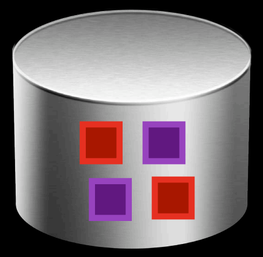
.Mac

Sync Client





Sync Client

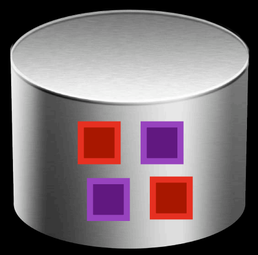


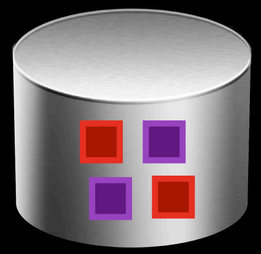
iSync

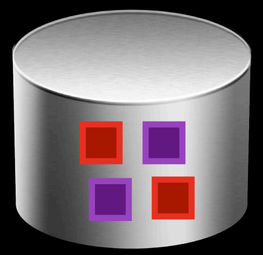
.Mac

Sync Client

Sync Client

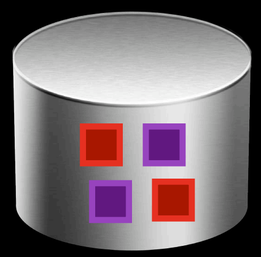






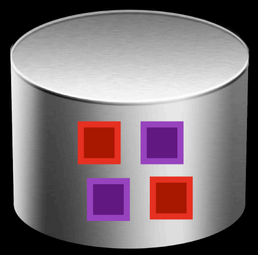
iSync

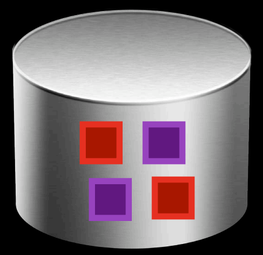
.Mac

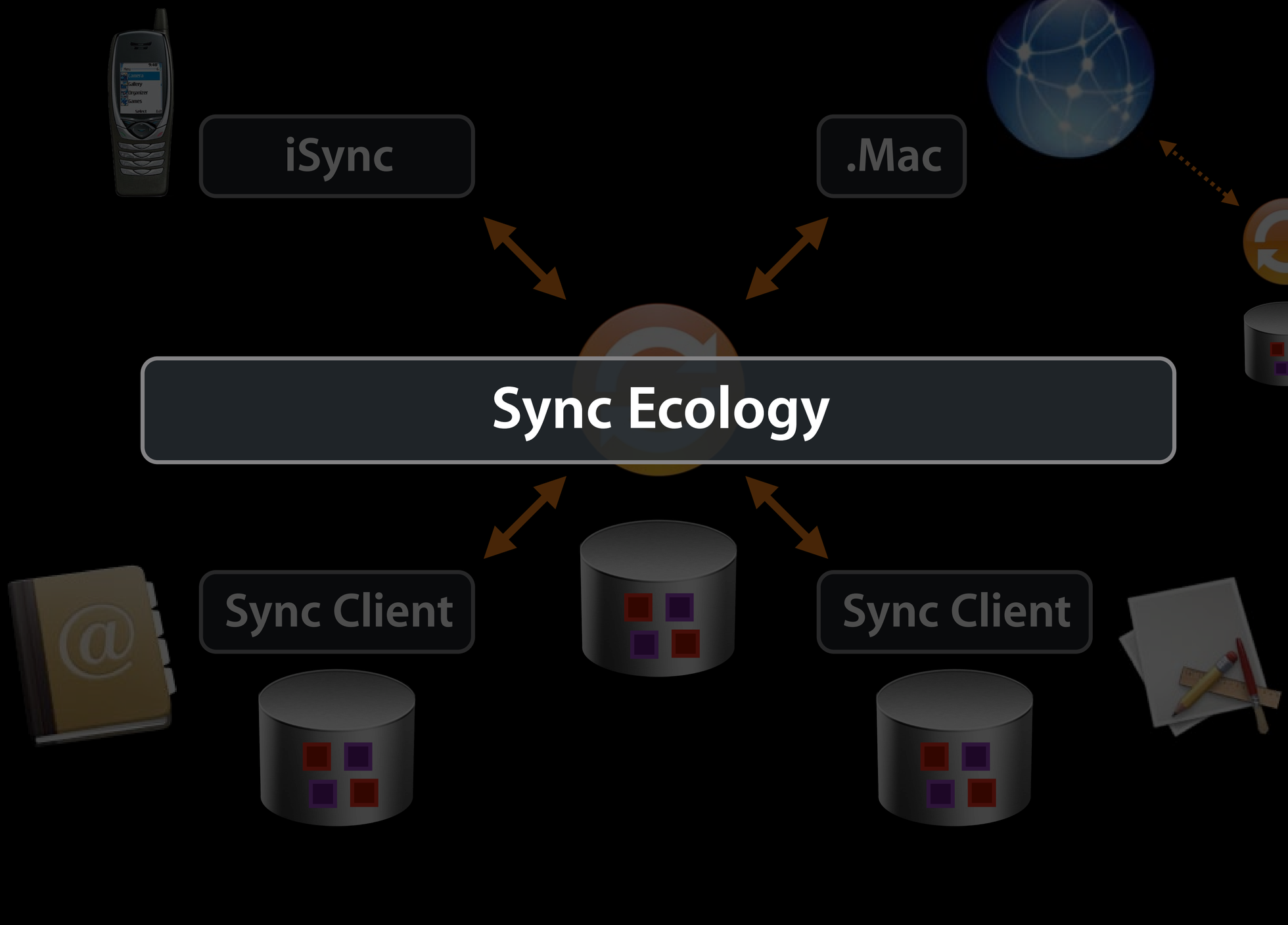


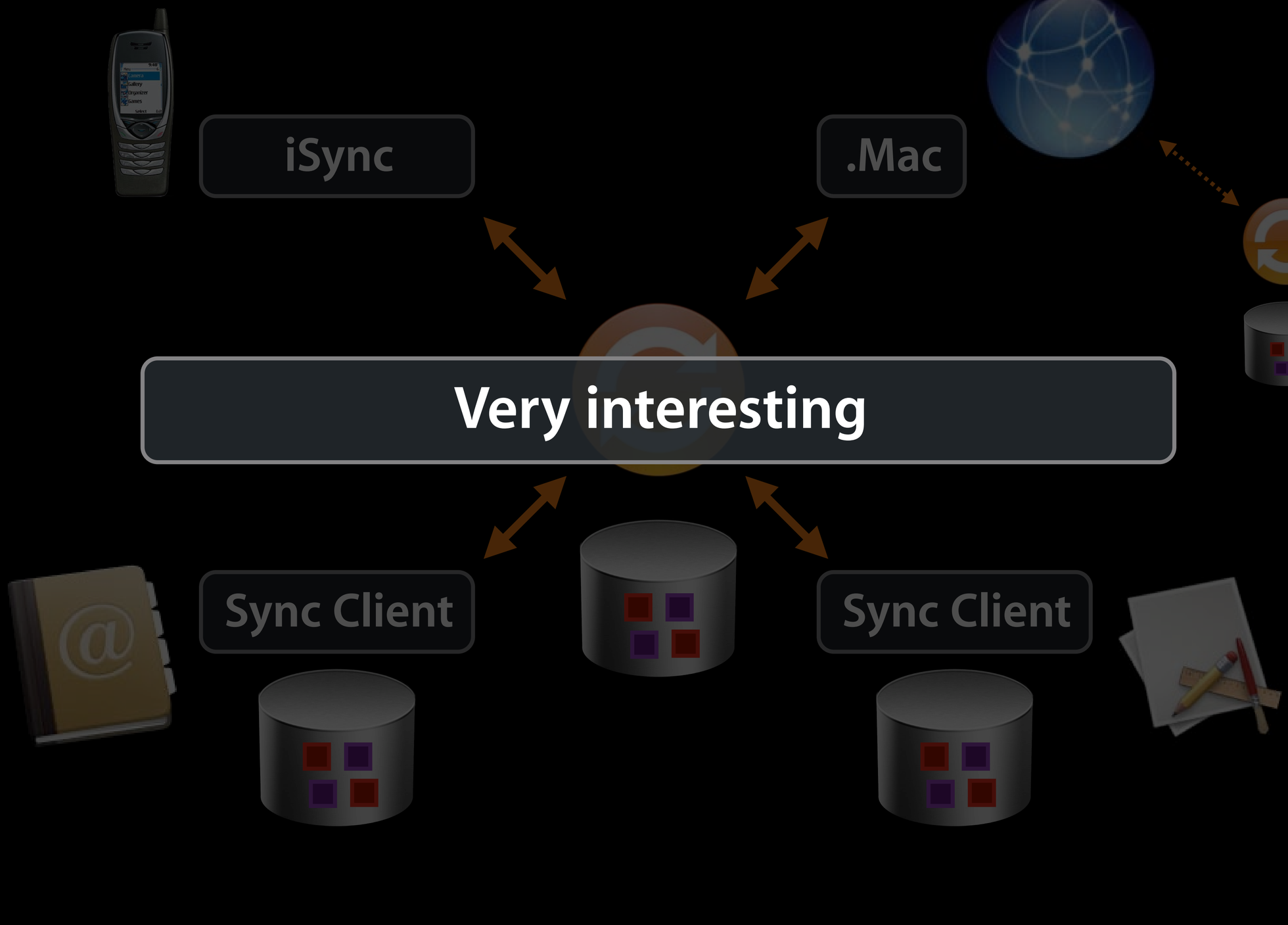
Sync Client

Sync Client





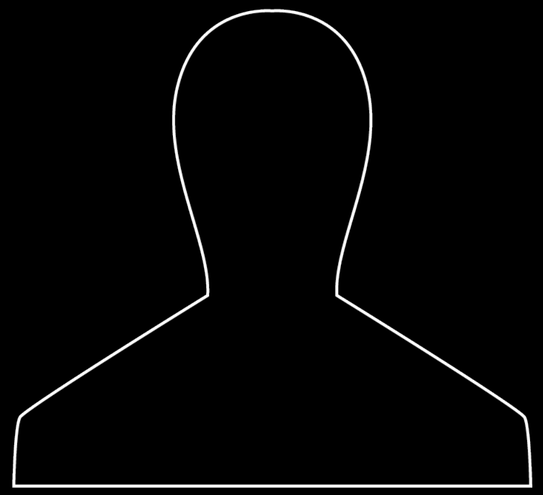

iSync
.Mac
Sync Ecology
Sync Client
Sync Client


iSync
.Mac
Very interesting
Sync Client
Sync Client

# SyncServices.framework

# Add Sync to your application!

# The People Application



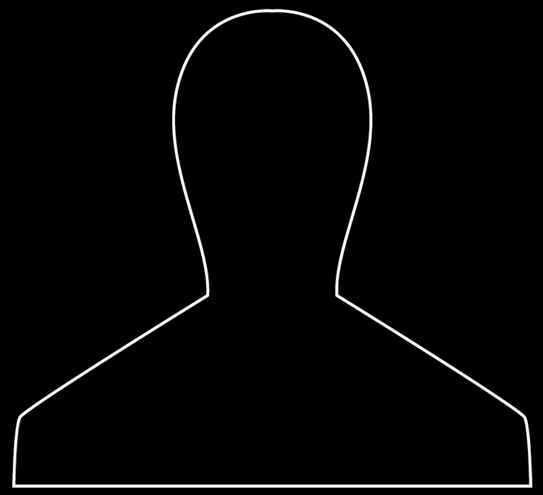
# A Sync Example

# Demo

## Syncing Contacts

http://developer.apple.com/wwdc2005/

# The People Application

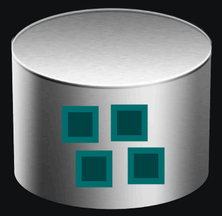


# A Sync Example

# Making Sync Work

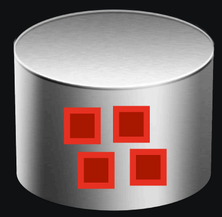
# Making Sync Work

## Application

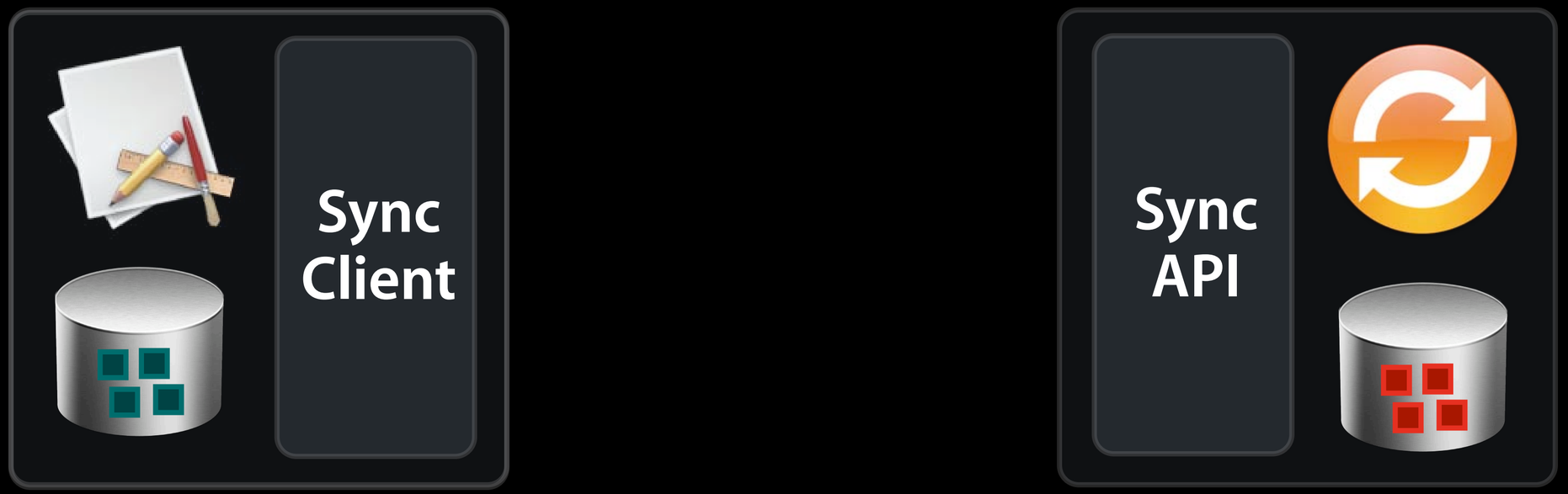


## Sync Engine

Sync API

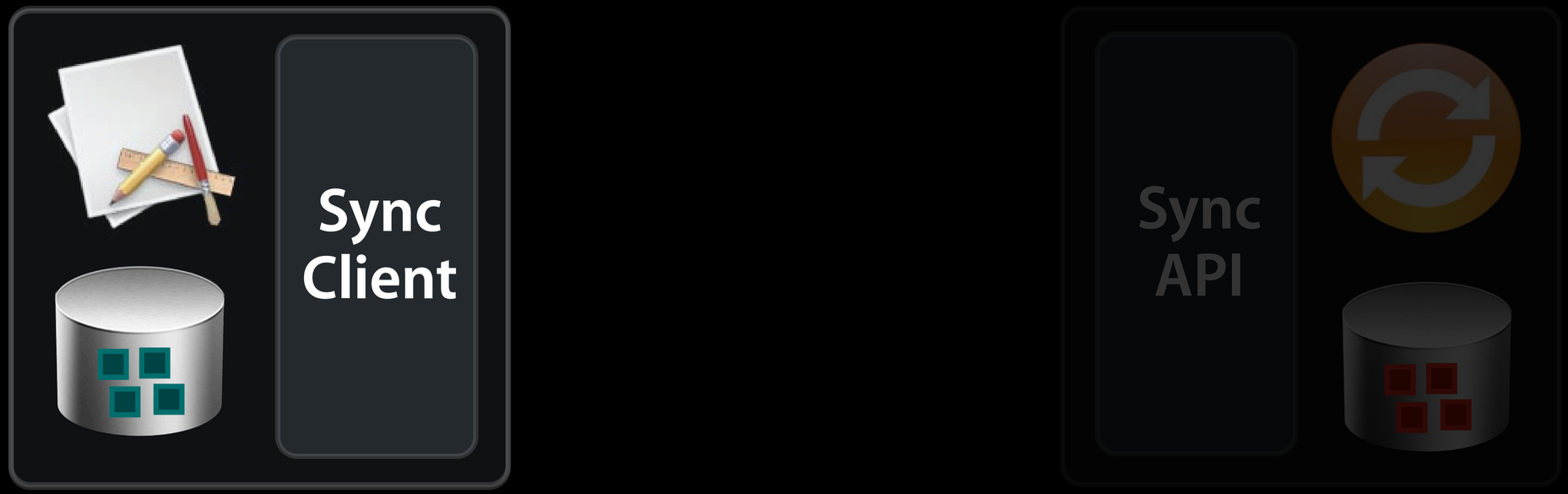


# Making Sync Work



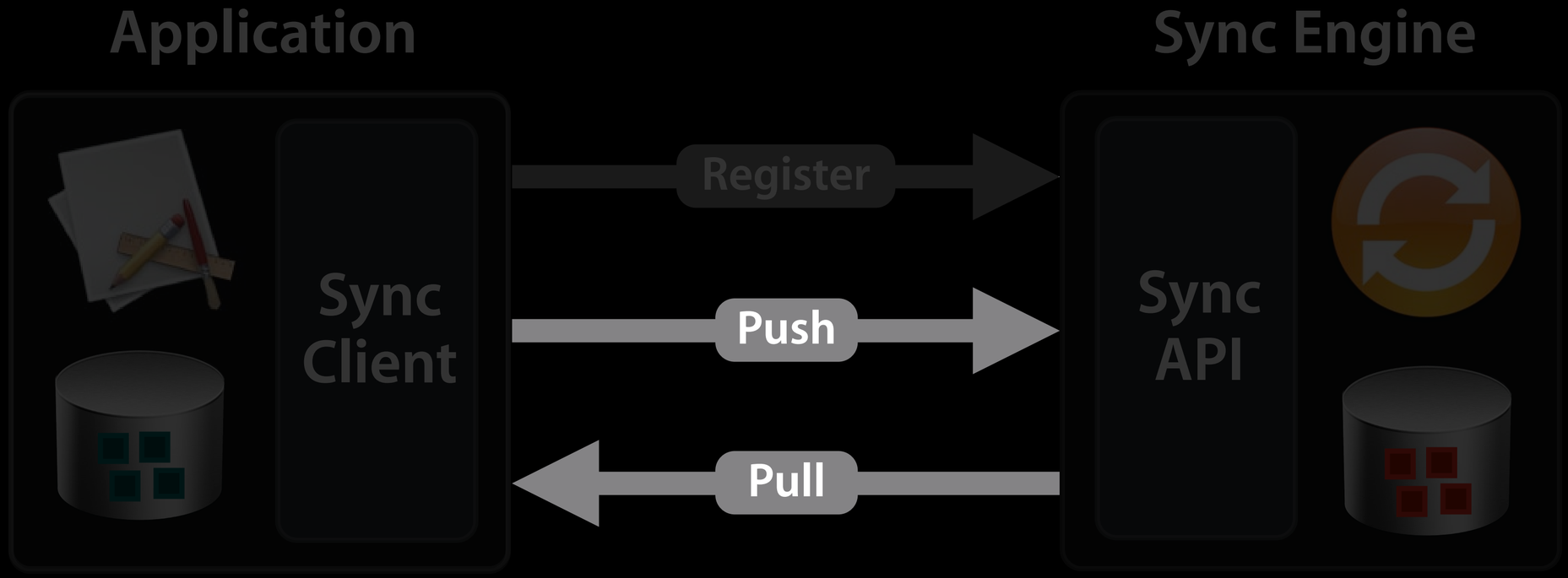
# Making Sync Work

## Data Handling



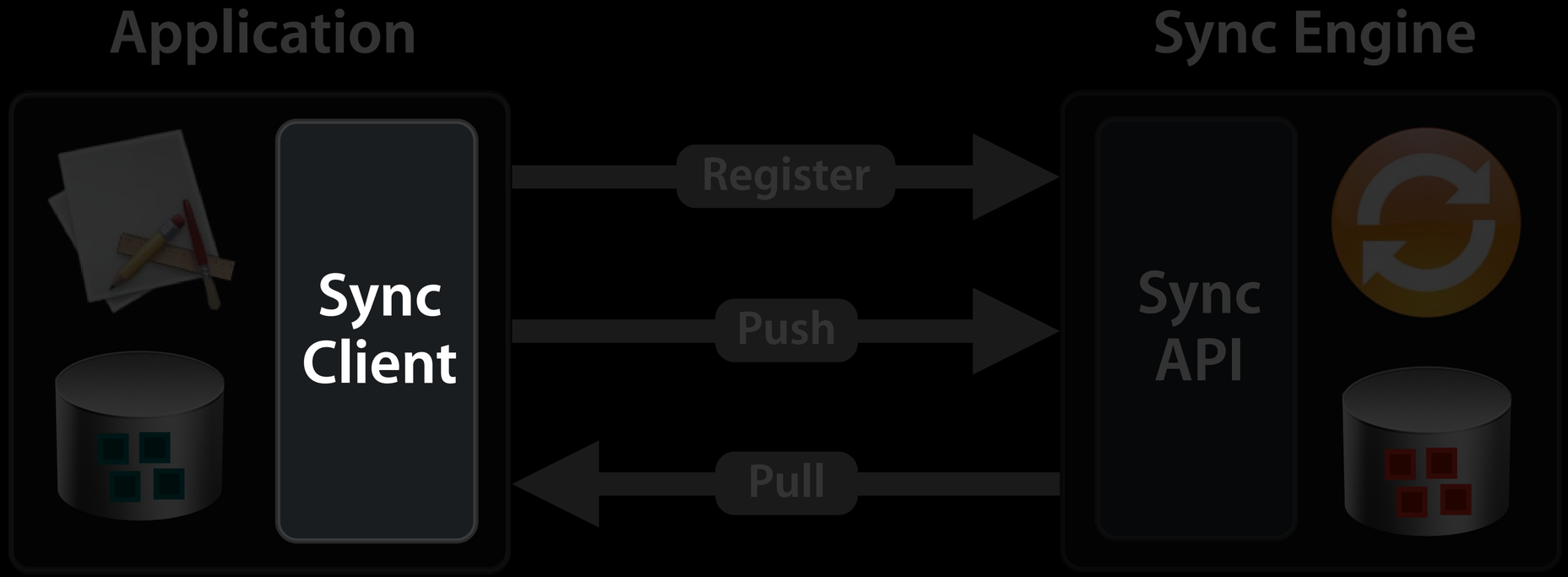
# Making Sync Work

# Making Sync Work




Session Control Flow

# Making Sync Work




Feature Decisions

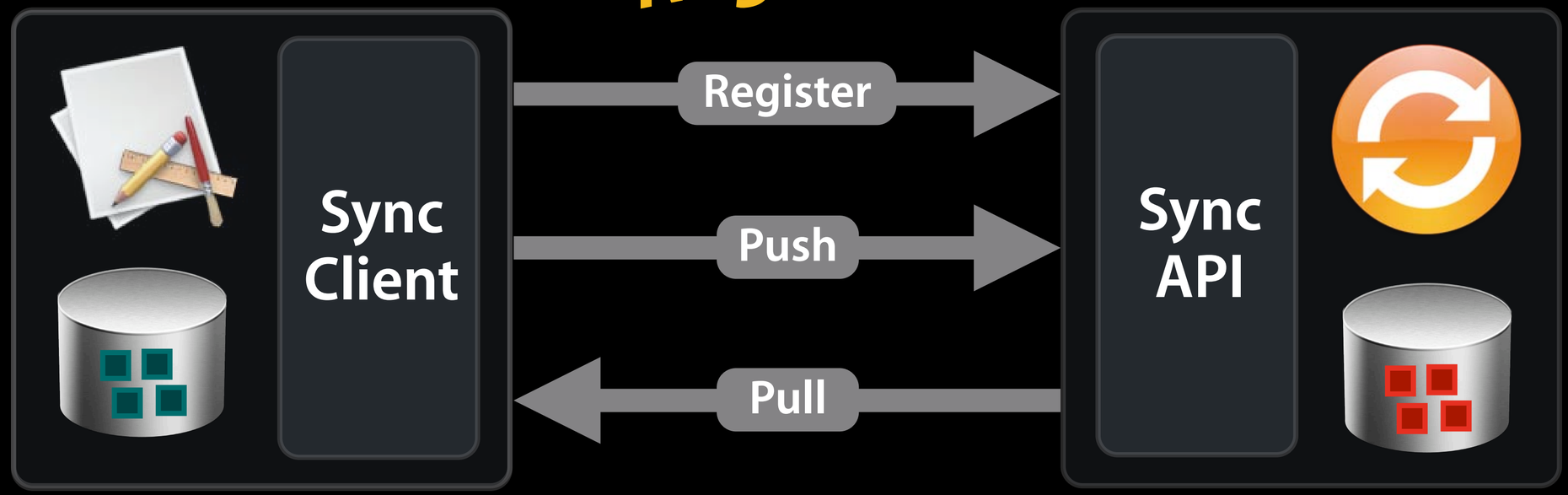
# Making Sync Work

Data Handling

Application

Registration

Sync Engine




Feature Decisions

Session Control Flow

# Hands-On Setup

# People.xcode

# FileName.m
## - doImportantWork:

```
- (NSDictionary *)convertAppRecordToSyncRecord:(NSDictionary *)record
{
    NSString *firstName = [record objectForKey:FirstNameKey];

    NSMutableDictionary *syncRecord = [NSMutableDictionary
       dictionaryWithObjectsAndKeys:
       EntityContact, ISyncRecordEntityNameKey, nil];

    if (firstName && ([firstName isEqualToString:@""] == NO)) {
       [syncRecord setObject:firstName forKey:FirstNameKey];
    }

    return syncRecord;
}
```

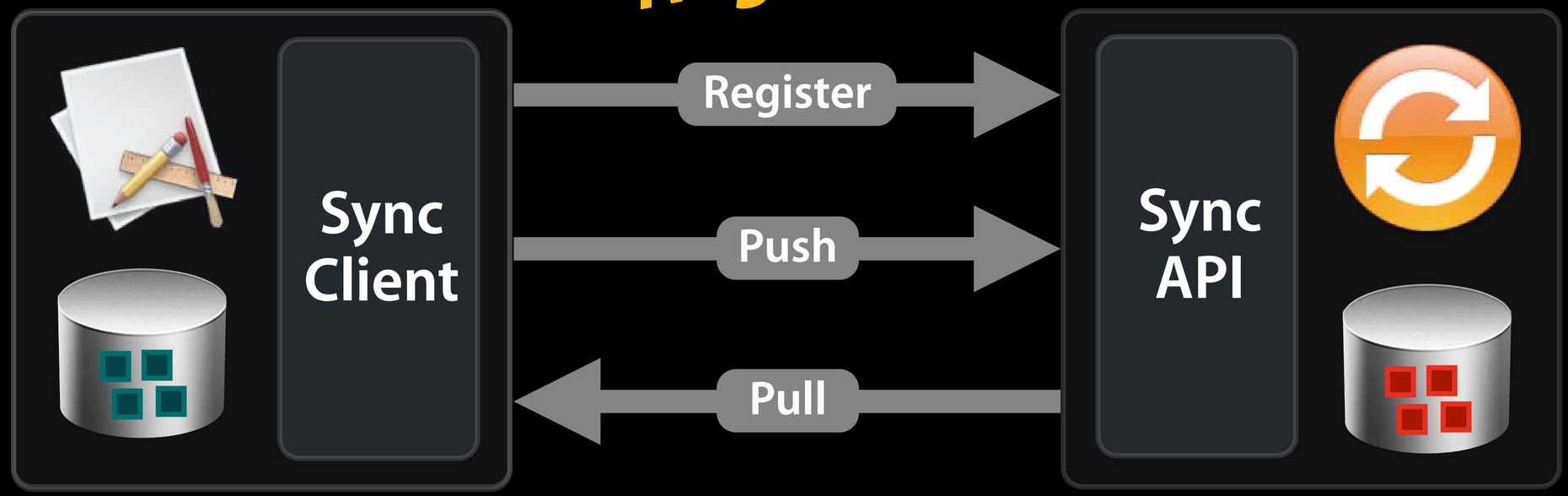
# Making Sync Work

Data Handling

Application

Registration

Sync Engine




Feature Decisions

Session Control Flow

# Making Sync Work

## Data Handling

Convert data to and from sync

Optimize: track data changes

# Making Sync Work

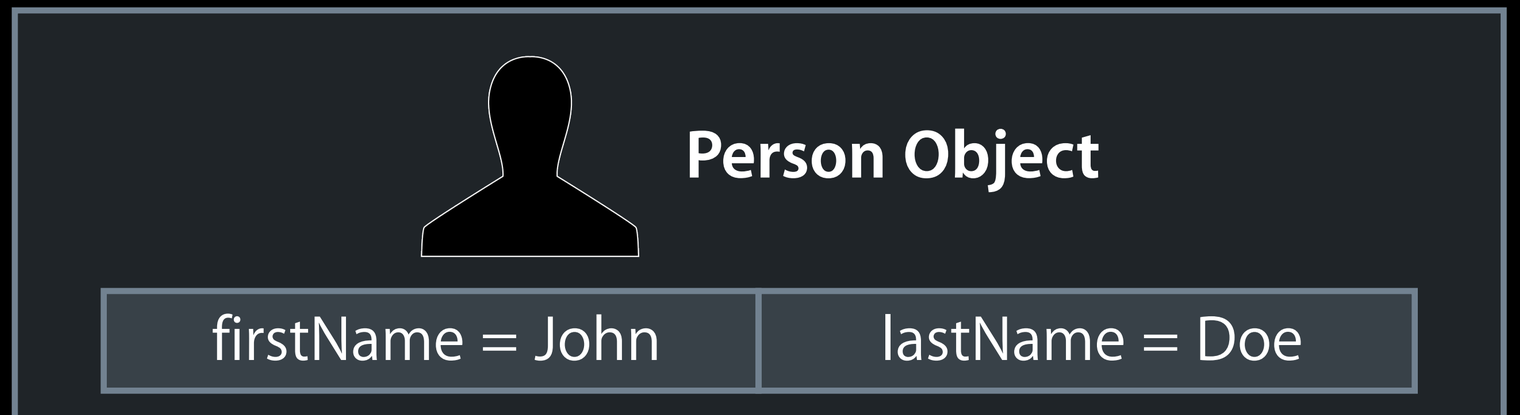
## Data Handling

Convert data to and from sync

Optimize: track data changes

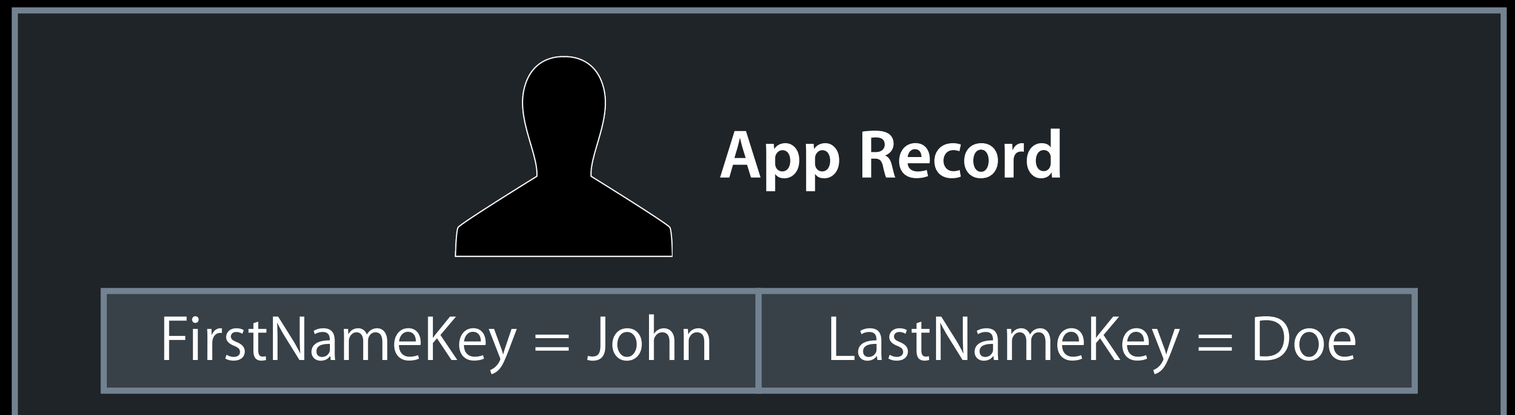
# Making Sync Work: Data Handling

## Convert data to and from sync



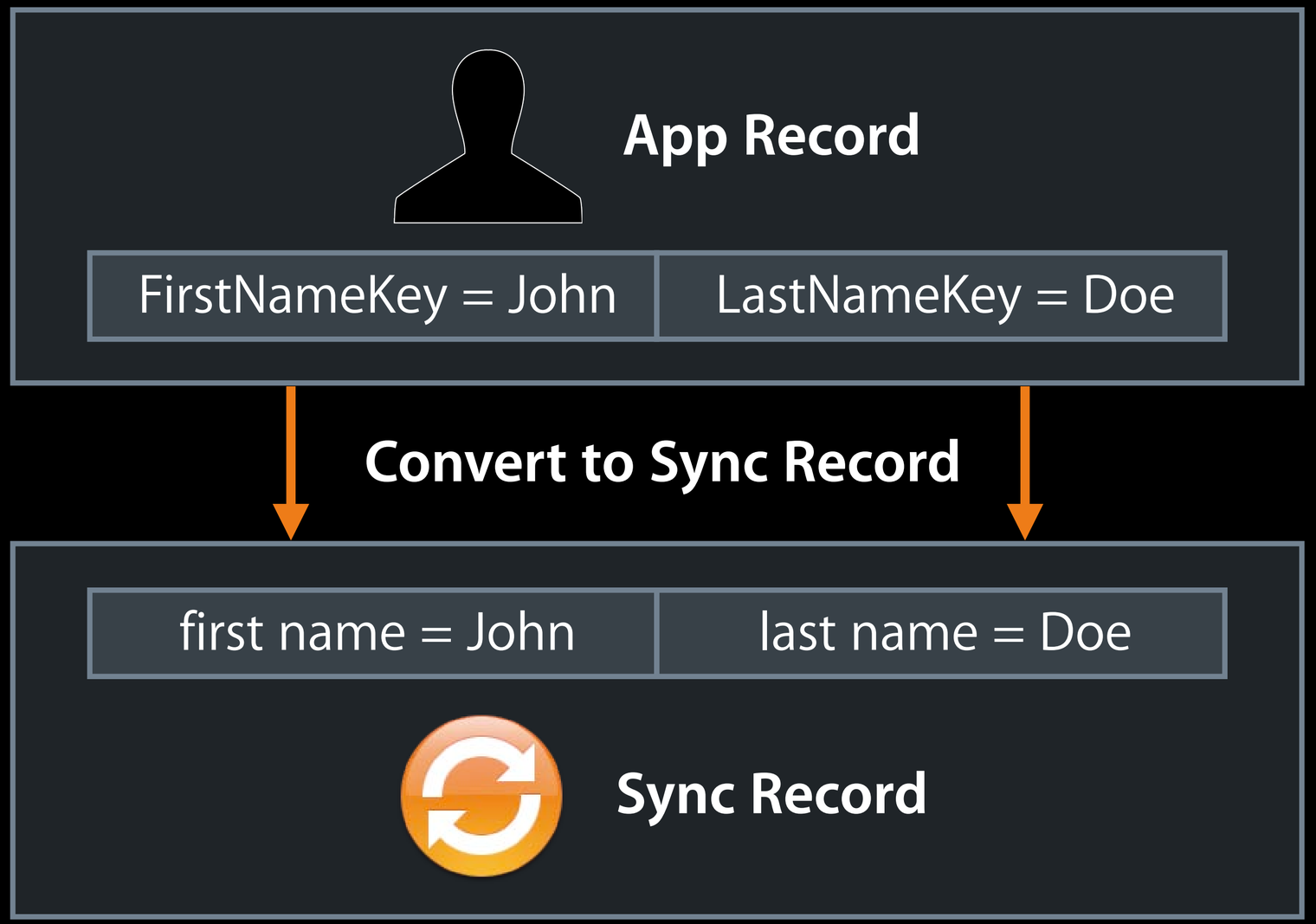
# Making Sync Work: Data Handling

## Convert data to and from sync



# Making Sync Work: Data Handling

## Convert data to and from sync



# AppControllerSyncing.m
## - convertAppRecordToSyncRecord:

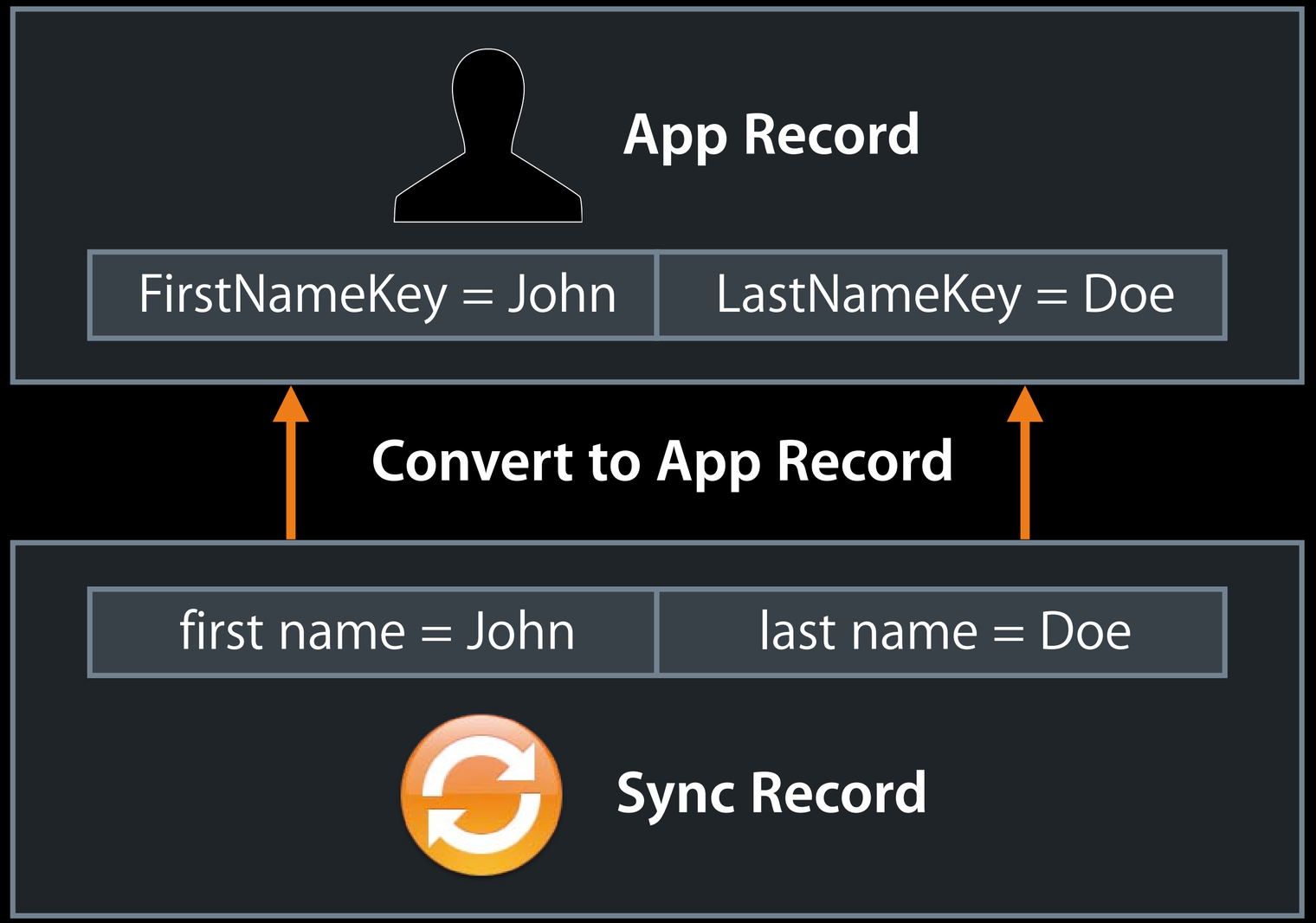
```
- (NSDictionary *)convertAppRecordToSyncRecord:(NSDictionary *)record
{
    NSString *firstName = [record objectForKey:FirstNameKey];
    NSMutableDictionary *syncRecord = [NSMutableDictionary
       dictionaryWithObjectsAndKeys:
       EntityContact, ISyncRecordEntityNameKey, nil];

    if (firstName && ([firstName isEqualToString:@""] == NO)) {
       [syncRecord setObject:firstName forKey:FirstNameKey];
    }
    ...

    return syncRecord;
}
```

# Making Sync Work: Data Handling

## Convert data to and from sync



# AppControllerSyncing.m

## - convertSyncRecordToAppRecord:withIdentifier:

```
- (NSDictionary *)convertSyncRecordToAppRecord:(NSDictionary *)record
      withIdentifier:(NSString *)identifier
{
      NSString *firstName = [record objectForKey:FirstNameKey];
      // formatting...
      NSDictionary *result = [NSDictionary dictionaryWithObjectsAndKeys:
         identifier, IdentifierKey,
         firstName ? firstName : @"", FirstNameKey,
         ... ,
         nil];
      // more formatting...

      return result;
}
```

# Making Sync Work

## Data Handling

Convert data to and from sync

Optimize: track data changes

# Making Sync Work

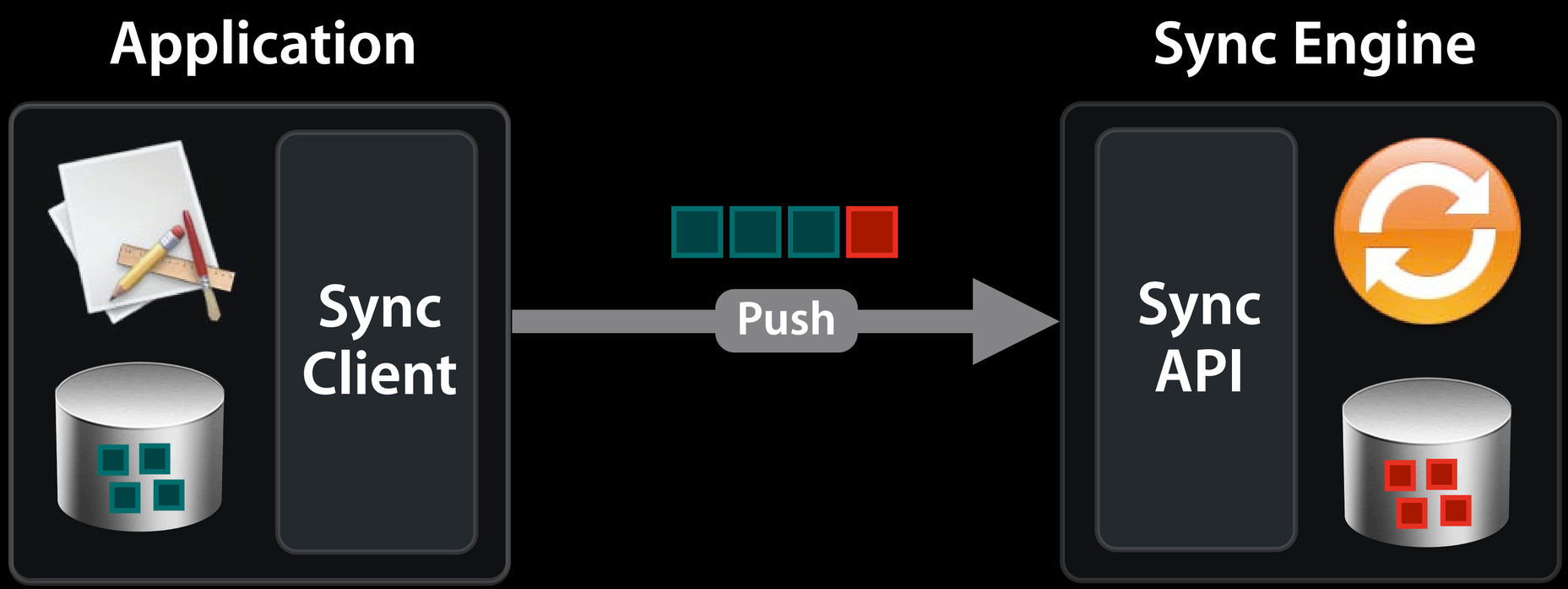
## Data Handling

Convert data to and from sync

Optimize: track data changes

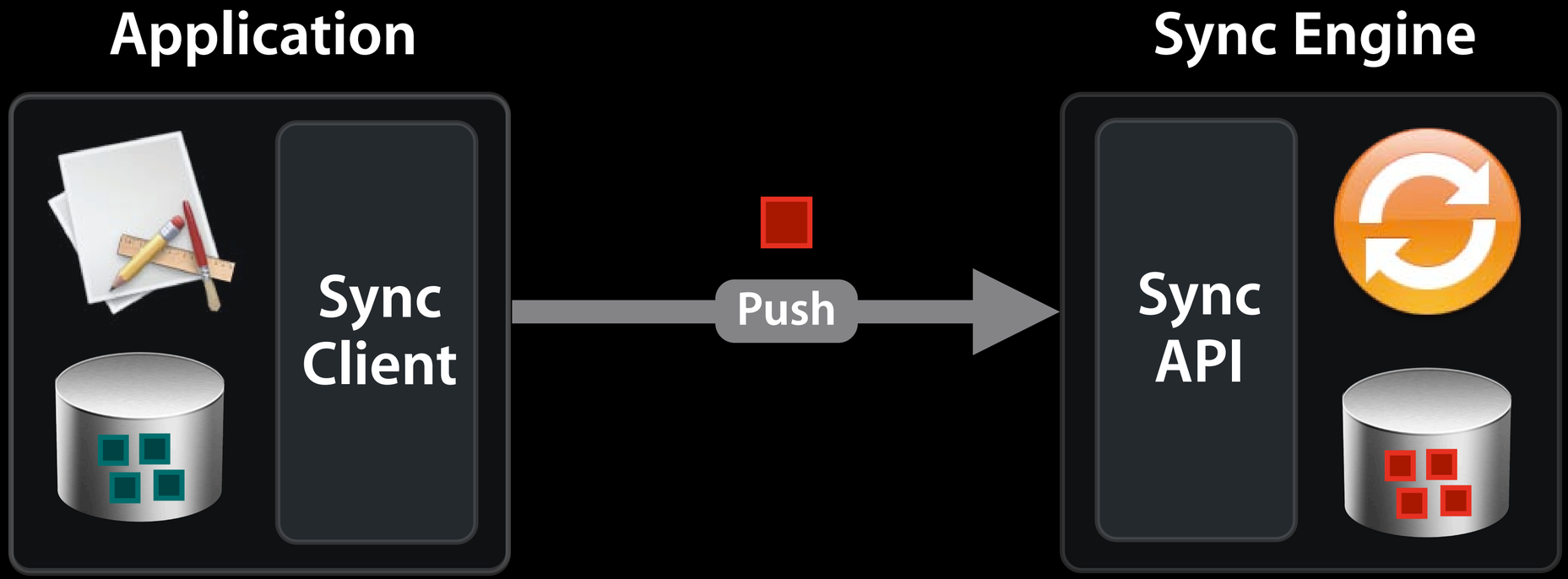
# Making Sync Work

## Optimize: track data changes



# Making Sync Work

## Optimize: track data changes



# Track Data Changes

## Strongly suggested

# Change.h

## Class Declaration

```
typedef enum {
	AddRecord,
	DeleteRecord,
	ModifyRecord,
} ChangeType;

@interface Change : NSObject
{
	ChangeType m_type;
	NSDictionary *m_record;
	NSDictionary *m_oldRecord;
	int m_row;
}
```

# AppController.m

## - tableView:setObjectValue:forTableColumn:row:

```
- (void)tableView:(NSTableView *)tableView setObjectValue:(id)object
    forTableColumn:(NSTableColumn *)tableColumn row:(int)row
{
    NSDictionary *oldRecord = [m_appRecords objectAtIndex:row];
    NSDictionary *record = nil;
    if ([[tableColumn identifier] isEqual:FirstNameKey]) {
       record = [NSDictionary dictionaryWithObjectsAndKeys:
       object, FirstNameKey, ...
       [oldRecord objectForKey:LastNameKey], LastNameKey,
       nil];
    }
    [self applyChange:[Change modifyRecordChange:record
       oldRecord:oldRecord row:row]];
}
```

# AppController.m

## - applyChange:

```
- (void)applyChange:(Change *)change
{
    switch ([change type]) {
       case AddRecord:    // add to record store
          break;
       case DeleteRecord: // delete from record store
          break;
       case ModifyRecord: // modify record store
          break;
    }
    [m_changes addObject:change];
    [[m_window undoManager] registerUndoWithTarget:self
       selector:@selector(unapplyChange:) object:change];
    // update user interface
}
```

# Making Sync Work

## Data Handling

Convert data to and from sync

Optimize: track data changes

# Making Sync Work

## Data Handling

Convert data to and from sync

Optimize: track data changes

# Making Sync Work

# Making Sync Work

Register Sync Schema

Register Sync Client Description

# Sync Schema

## Describes data to the Sync Engine

# Entity

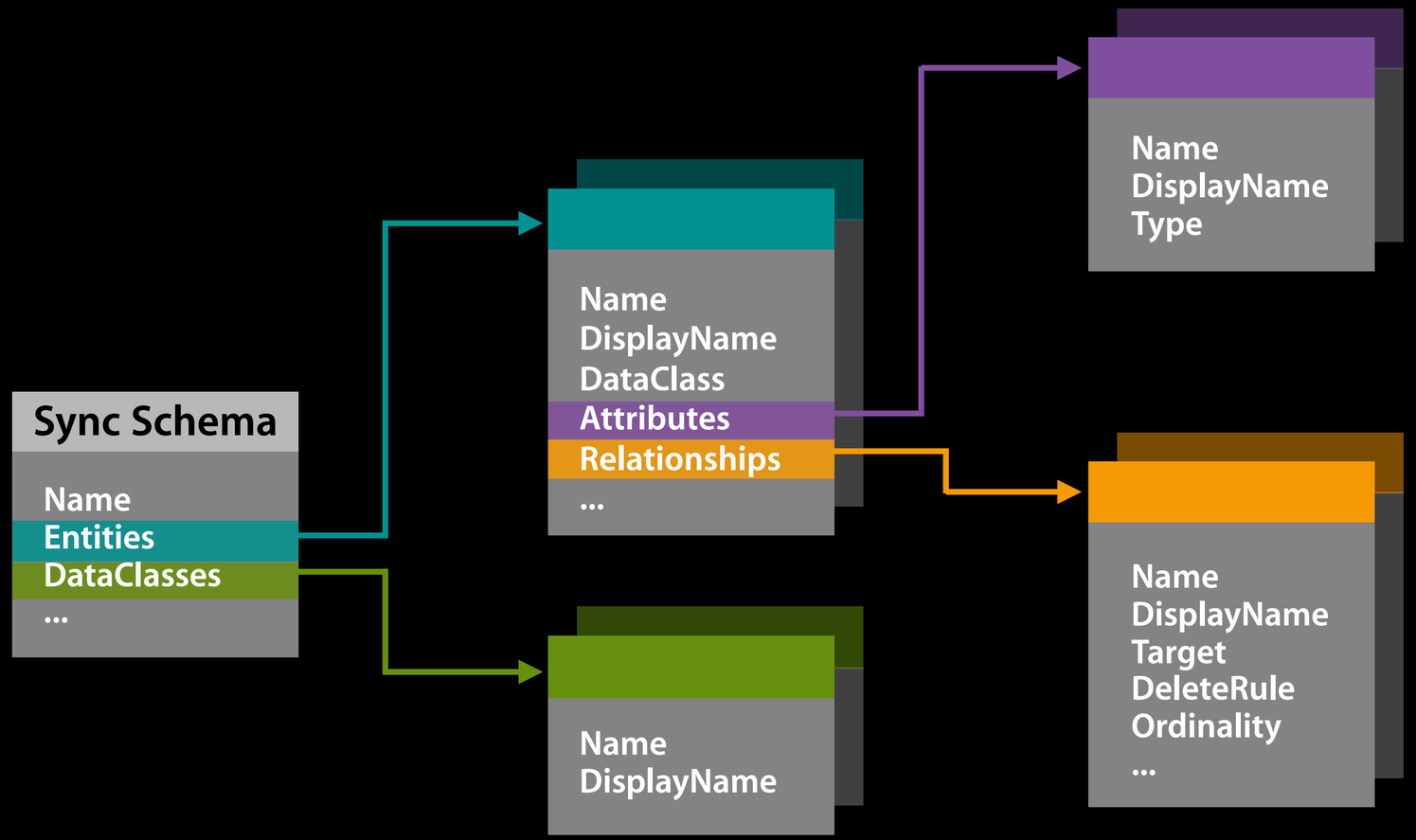
## Element of a data model

# Relationship

## Reference to another Entity

# Making Sync Work: Registration

## Sync Schema

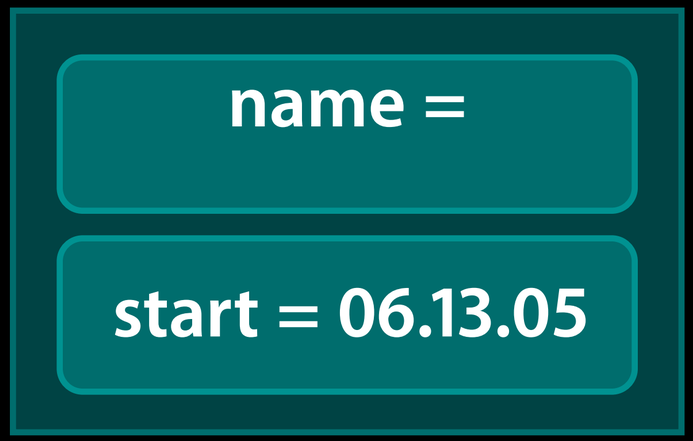


# Data Class

## Logical group of Entities

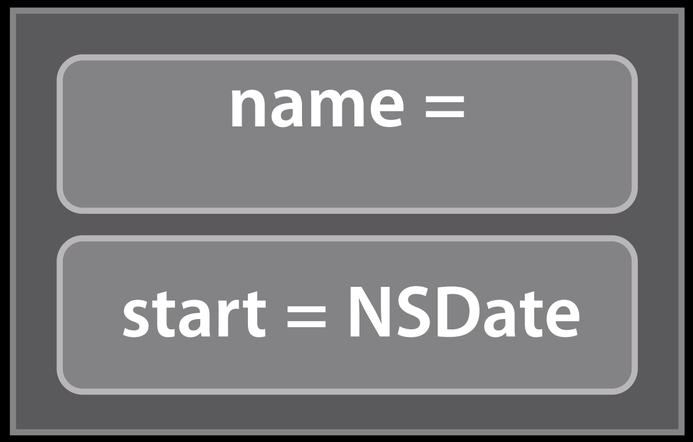
# Making Sync Work: Registration

## Schema example



# Making Sync Work: Registration

## Schema example



# Making Sync Work: Registration

## Use an existing Schema

Contacts

Calendars

Bookmarks

# Making Sync Work: Registration

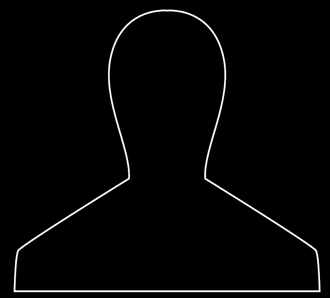
## What's in the Contacts Schema?

**Contacts**

| | | |
|---|---|---|
| Contact | Email Addresses | Relations |
| Group | Street Addresses | IM |
| Phone Numbers | Birthdays | URLs |

# Making Sync Work: Registration

## Sync with part of a Schema



**Contact**

- First Name
- Middle Name
- Last Name
- Company Name

**Contact**

- First Name
- Middle Name
- Last Name
- Company Name
- Hobby
- Birthday
- ... and more...

# Sync Schema

## Described in a .plist file

# Sync Schema

## Make one in Xcode

# Making Sync Work

Register Sync Schema

Register Sync Client Description

# Sync Client Description

## Tells Sync Engine about your Sync Client

# Sync Client Description

## Described in a .plist file

# Sync Schema

## Describes data to the Sync Engine

# Sync Client Description

## Bridge between application and Sync Engine

# Sync Schema

## What can possibly be synced

# Sync Client Description

## What I want to sync in this application

# Sync Schema

## Described in a .plist file

# Sync Client Description

## Described in a .plist file

# Sync Schema

## MySchema.plist

# Sync Client Description

## ClientDescription.plist

# MySchema.plist

## Schema Structure Overview

```
<?xml version="1.0" encoding="UTF-8"?> ...
<dict>
  <key>DataClasses</key><array>...</array>
  <key>Entities</key>
     <dict>
        <key>DataClass</key><string>com.mycompany.Schema</string>
        <key>Name</key><string>com.mycompany.Schema.Entity</string>
        <key>Attributes</key><array>...</array>
        <key>Relationships</key><array>...</array>
        <key>IdentityProperties</key><array>...</array>
     </dict>
</dict>
...
```

# ClientDescription.plist

## Describe Sync Client

```
<?xml version="1.0" encoding="UTF-8"?> ...
<dict>
  <key>Entities</key>
  <dict>
     <key>com.apple.contacts.Contact</key>
     <array>
        <string>com.apple.syncservices.RecordEntityName</string>
        <string>first name</string>
        <string>middle name</string>
        <string>last name</string>
        <string>company name</string>
     </array>
  </dict>
</dict>
...
```

# AppControllerSyncing.m

## - performRegistrations

```
- (ISyncClient *)performRegistrations
{
    // register a schema

    ISyncManager *syncManager = [ISyncManager sharedManager];
    [syncManager registerSchemaWithBundlePath:SchemaPath];

    ...
}
```

# AppControllerSyncing.m

## - performRegistrations

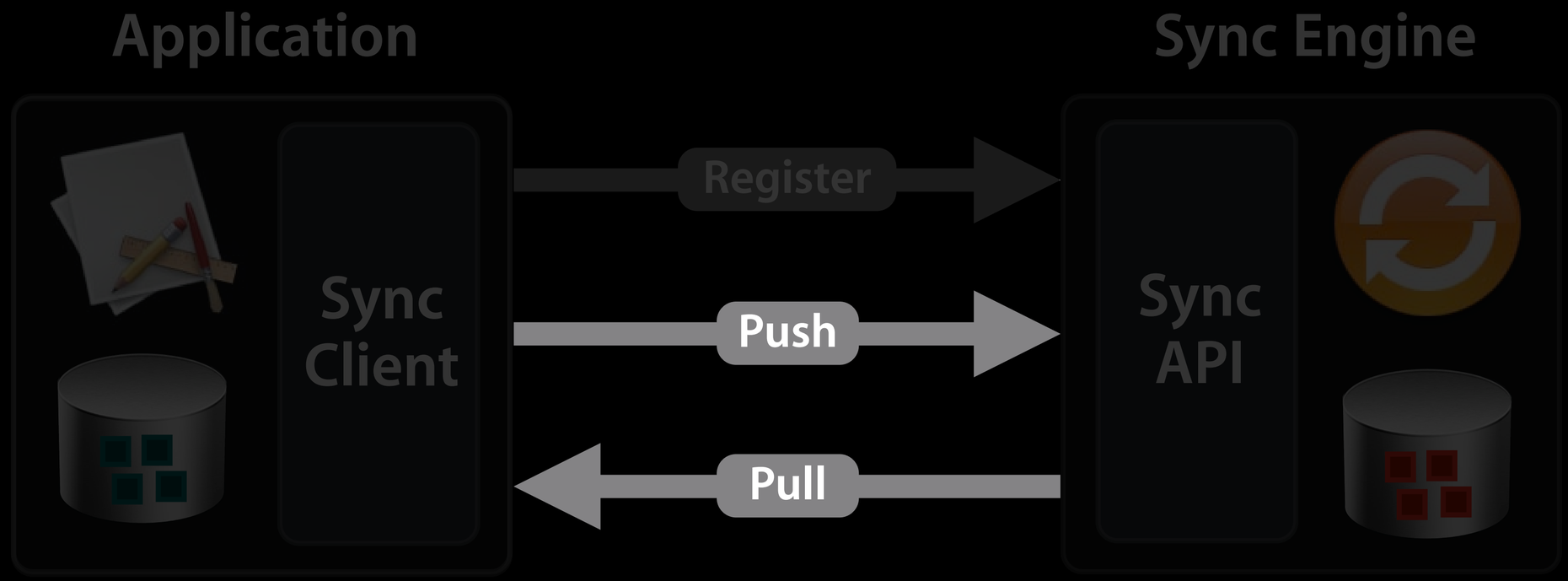
```
- (ISyncClient *)performRegistrations
{
    // register a client
    ISyncManager *syncManager = [ISyncManager sharedManager];
    ...
    ISyncClient *syncClient;
     if (!(syncClient = [syncManager
          clientWithIdentifier:ClientIdentifier])) {
        syncClient = [syncManager
          registerClientWithIdentifier:ClientIdentifier
          descriptionFilePath:[[NSBundle mainBundle]
          pathForResource:@"ClientDescription" ofType:@"plist"]];
    }
    return syncClient;
}
```

# Making Sync Work

Register Sync Schema

Register Sync Client Description

# Making Sync Work




Session Control Flow

# Making Sync Work

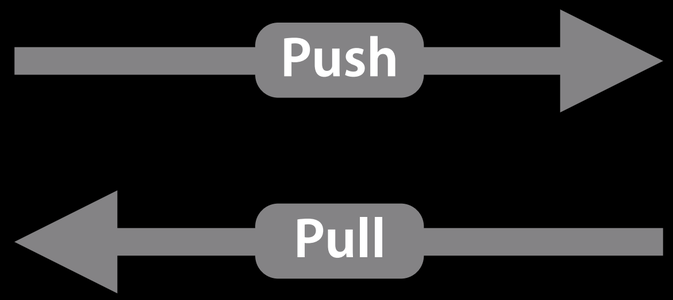
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Making Sync Work

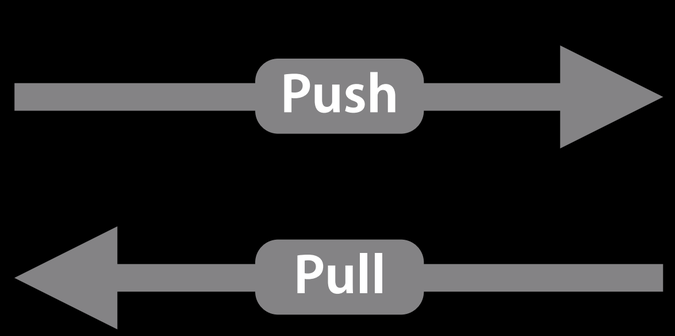
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Sync Session

## One complete transaction with Sync Engine

# AppControllerSyncing.m

## - sync:

```
// synchronous

ISyncSession *session = [ISyncSession beginSessionWithClient:syncClient
     entityNames:array beforeDate:timeoutDate];

...

// asynchronous
- (IBAction)sync:(id)sender {
     ...
     [ISyncSession beginSessionInBackgroundWithClient:syncClient
        entityNames:array
        target:self selector:@selector(performSync::)];
     ...
}
```

# AppControllerSyncing.m

## - performSync::

```
- (void)performSync:(ISyncClient *)syncClient
                   :(ISyncSession *)syncSession
{
     @try {
        // Sync session could be nil.
        if (syncSession) {
            [self negotiateSession:syncSession];
            [self pushDataForSession:syncSession];
            [self pullDataForSession:syncSession];
        }
     }
     @catch ...
}
```

# Making Sync Work

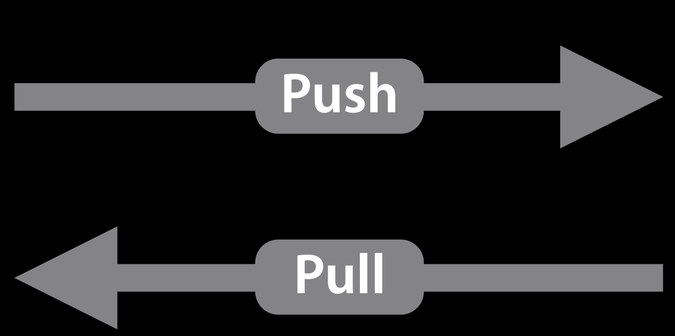
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Making Sync Work

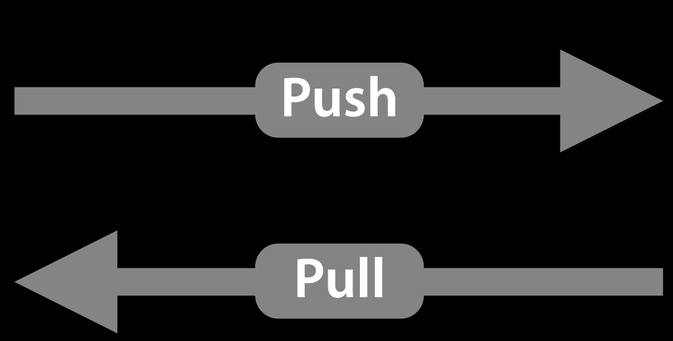
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Negotiate Session

## Ask: but you may not get what you ask for

# Negotiate Session

Fast  Slow  Refresh  Pull The Truth

# Negotiate Session

## Fast:  I will tell you about my changes

# Negotiate Session

## Slow:  I will tell you about all my records

# Negotiate Session

## Refresh:  I will tell you about the records I have

# Negotiate Session

## Pull The Truth:  I want to replace all my records

# AppControllerSyncing.m

## - negotiateSession:

```
- (void)negotiateSession:(ISyncSession *)syncSession {
    switch (desiredSyncMode) {
        case FastSync: // Fast sync is the default.
            break;
        case SlowSync:
            [syncSession clientWantsToPushAllRecordsForEntityNames:array];
            break;
        case RefreshSync:
            [syncSession clientDidResetEntityNames:array];
            break;
        case PullTheTruth:
            // Must be handled before syncSession starts.
            break;
    }
}
```

# AppControllerSyncing.m

## - sync:

```
- (IBAction)sync:(id)sender
{
    ...

    // If you are pulling the truth, tell the client this fact
    // before starting the session.
    if (desiredMode == PullTheTruth) {
       [syncClient setShouldReplaceClientRecords:YES
          forEntityNames:array];
    }

    [ISyncSession beginSessionInBackgroundWithClient:...];

    ...
}
```

# Making Sync Work

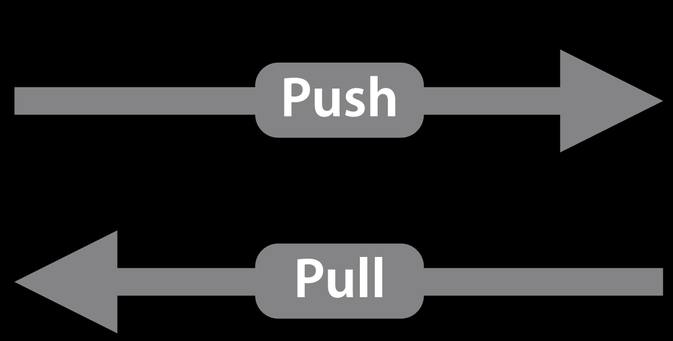
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Making Sync Work

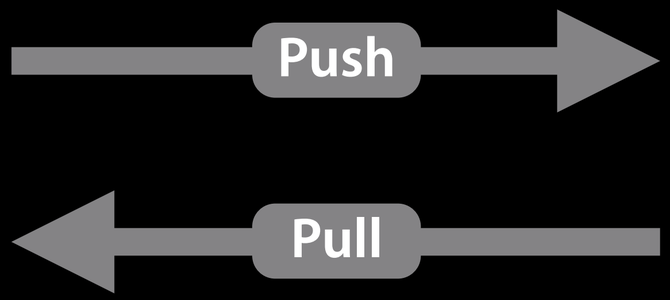
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Push

## Tell the Sync Engine about application data

# AppControllerSyncing.m

## - pushDataForSession:

```
- (void)pushDataForSession:(ISyncSession *)syncSession
{
   if ([syncSession shouldPushAllRecordsForEntityName:EntityContact]) {
      NSEnumerator *enumerator = [records objectEnumerator];
      NSDictionary *appRecord;
      while ((appRecord = [enumerator nextObject])) {
         NSDictionary *syncRecord =
            [self convertAppRecordToSyncRecord:appRecord];
         NSString *identifier = [appRecord objectForKey:IdentifierKey];
         [syncSession pushChangesFromRecord:syncRecord
            withIdentifier:identifier];
      }
   }
   ...
}
```

# AppControllerSyncing.m

## - pushDataForSession:

```
- (void)pushDataForSession:(ISyncSession *)syncSession {
   ...
   else if ([syncSession shouldPushChangesForEntityName:EntityContact]) {
      NSEnumerator *enumerator = [m_changes objectEnumerator];
      Change *change;
      while ((change = [enumerator nextObject])) {
         switch ([change type]) {
            case AddRecord:
            case ModifyRecord:
            case DeleteRecord:
               // handle change
               break;
         }
   }
   ...
```

# AppControllerSyncing.m

## - pushDataForSession:

```
- (void)pushDataForSession:(ISyncSession *)syncSession {
    switch ([change type]) {
        case AddRecord:
        case ModifyRecord:
            NSDictionary *appRecord = [change record];
            NSString *identifier = [appRecord objectForKey:IdentifierKey];
            NSDictionary *syncRecord = [self
                convertAppRecordToSyncRecord:appRecord];
            [syncSession pushChangesFromRecord:syncRecord
                withIdentifier:identifier];
            // could do even better...
            // [syncSession pushChange:[change mySyncChangeWithChange]];
            break;
    }
...
```

# Making Sync Work

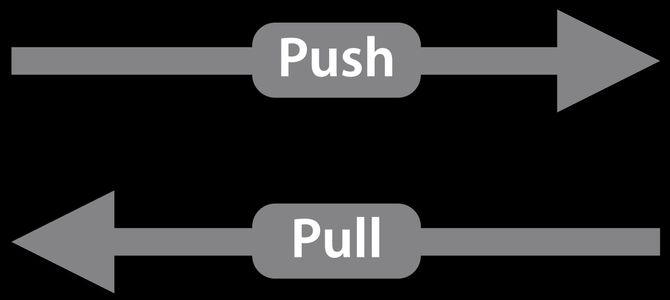
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Making Sync Work

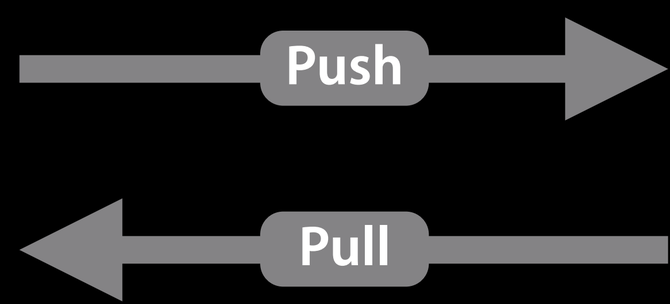
Start Session

Negotiate

Push

**Pull**

Finish or Cancel




Session Control Flow

# Pull

## Get changes from the Sync Engine

# AppControllerSyncing.m

## - pullDataForSession:

```
- (void)pullDataForSession:(ISyncSession *)syncSession
{
   if (![syncSession shouldPullChangesForEntityName:entity])
      // no pulling to be done
      return;

   ...
```

# AppControllerSyncing.m

## - pullDataForSession:

```
- (void)pullDataForSession:(ISyncSession *)syncSession
{
   ...
   if ([syncSession shouldReplaceAllRecordsOnClientForEntityName:entity])
      // pulling the truth
      replaceAllRecords = YES;

   if (![syncSession prepareToPullChangesForEntityNames:array
      beforeDate:[NSDate distantFuture]])
      return; // failed

   if (replaceAllRecords)
      // OK to delete your data
      [m_appRecordsForSession removeAllObjects];
   ...
```

# AppControllerSyncing.m

## - pullDataForSession:

```
- (void)pullDataForSession:(ISyncSession *)syncSession
{
   ...
   NSEnumerator *changeEnumerator =
      [syncSession changeEnumeratorForEntityNames:entityNames];
   ISyncChange *change;
   while ((change = [changeEnumerator nextObject])) {
      switch ([change type]) {
         case ISyncChangeTypeAdd:     // convert record and add
         case ISyncChangeTypeModify:  // convert record and replace
         case ISyncChangeTypeDelete:  // just delete
            break;
      }
   }
   ...
```

# Making Sync Work

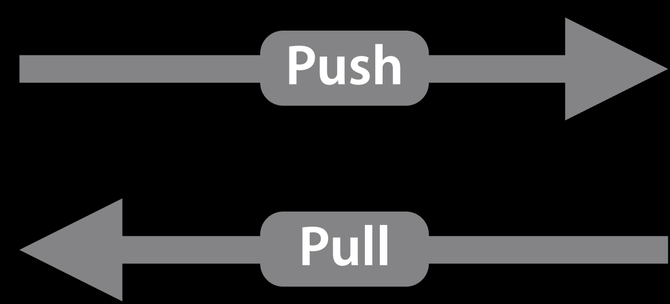
Start Session

Negotiate

Push

**Pull**

Finish or Cancel




Session Control Flow

# Making Sync Work

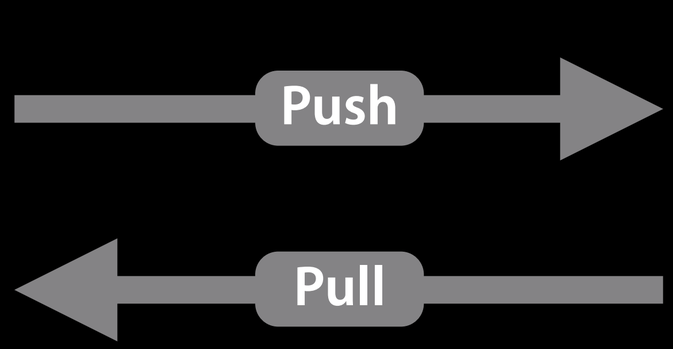
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Cancel

## User cancel… or something went wrong

# AppControllerSyncing.m

## - syncFailed:

```
- (void)syncFailed:(ISyncSession *)syncSession error:(NSError *)error
{
    [syncSession cancelSyncing];
    NSLog(@"sync failed: %@", [error localizedFailureReason]);
    [self syncCleanup];
}
```

# Finish

## Two-part process

# AppControllerSyncing.m

## - pullDataForSession:

```
- (void)pullDataForSession:(ISyncSession *)syncSession
{
   ...
   // First part of commit.
   NSString *identifier;
   NSEnumerator *enumerator = [pulledIdentifiers objectEnumerator];
   while ((identifier = [enumerator nextObject])) {
      [syncSession clientAcceptedChangesForRecordWithIdentifier:identifier
         formattedRecord:nil newRecordIdentifier:nil];
   }
   // Second part of commit.
   [syncSession clientCommittedAcceptedChanges];
   [syncSession finishSyncing];
   ...
}
```

# Making Sync Work

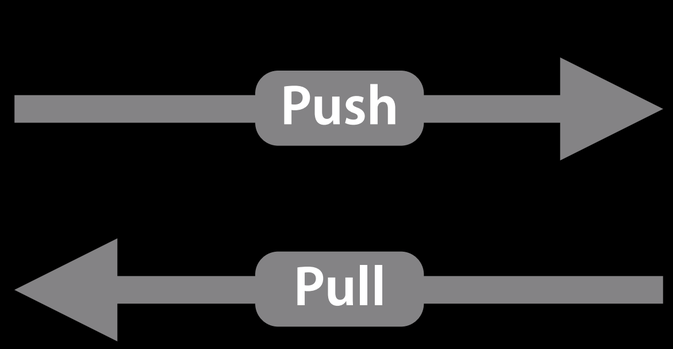
Start Session

Negotiate

Push

Pull

Finish or Cancel




Session Control Flow

# Making Sync Work

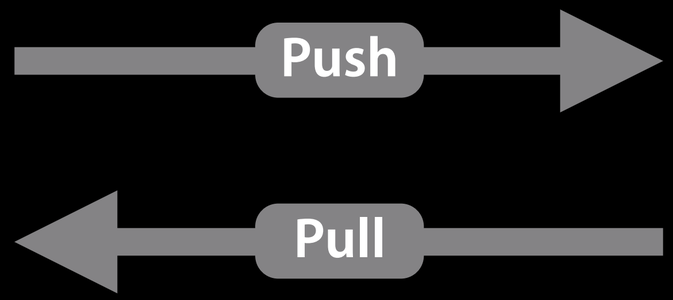
Start Session

Negotiate

Push

Pull

Finish or Cancel



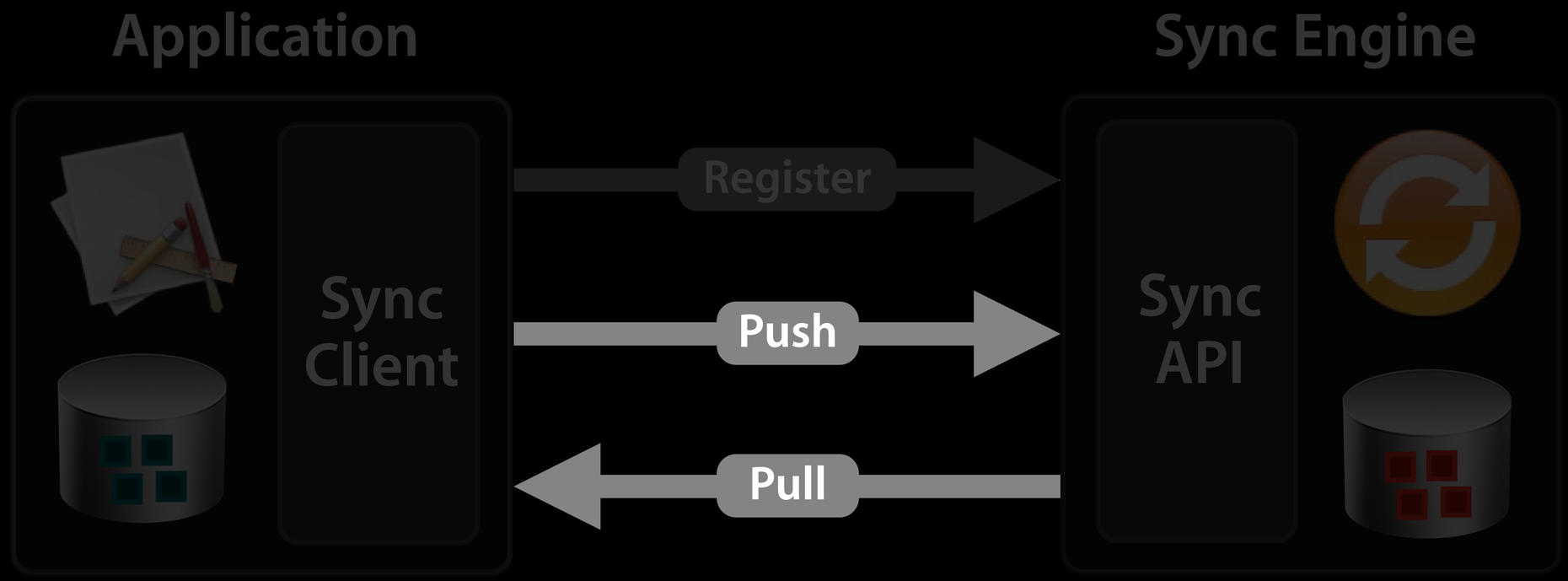

Session Control Flow

# Demo

## Session Control Flow

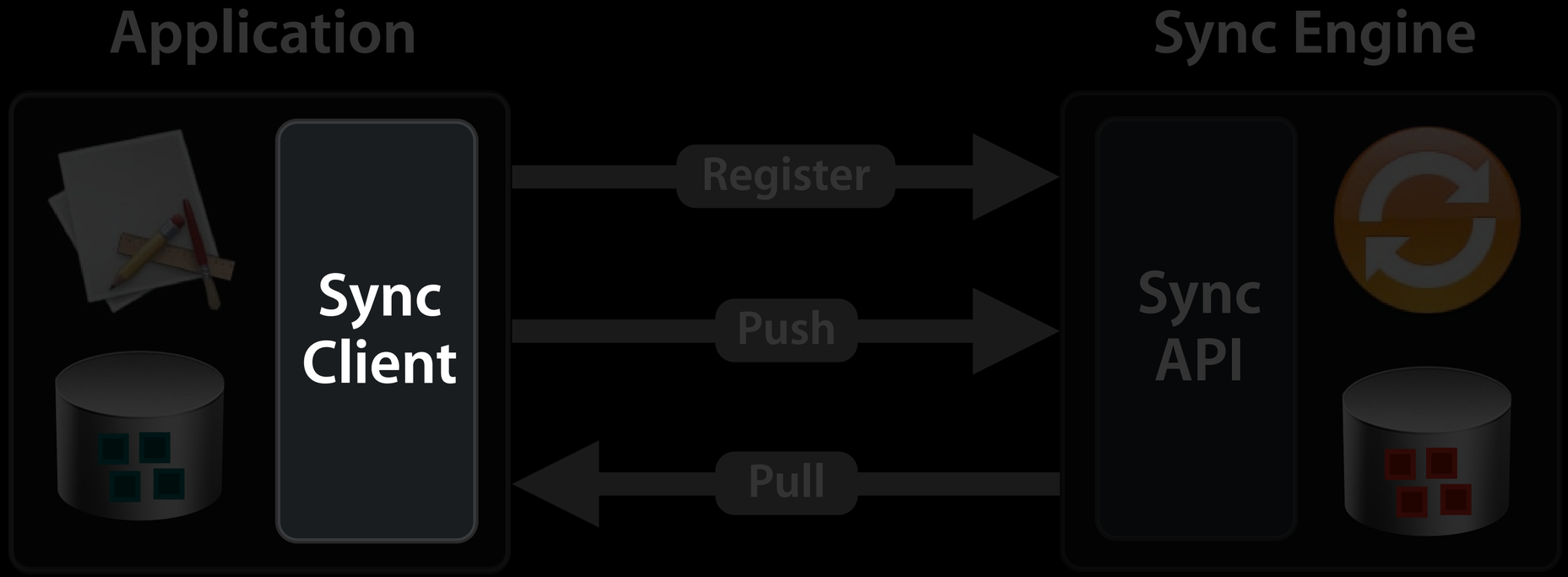
http://developer.apple.com/wwdc2005/

# Making Sync Work




Session Control Flow

# Making Sync Work




Feature Decisions

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

*Feature Decisions*

# Making Sync Work

Sync Client

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Record Filtering

## Ignore records

# LastNameFilter.h

## Interface

```
@interface LastNameFilter: NSObject <NSCoding, ISyncFiltering>
+ (LastNameFilter *)filter;
@end
```

# LastNameFilter.m

## Implementation

```
- (id)initWithCoder:(NSCoder *)coder { ... }
- (void)encodeWithCoder:(NSCoder *)coder { ... }
- (BOOL)isEqual:(id)other { ... }

- (NSArray *)supportedEntityNames
{
    return [[AppController sharedAppController] entityNames];
}

- (BOOL)shouldApplyRecord:(NSDictionary *)record
     withRecordIdentifier:(NSString *)recordIdentifier
{
    NSString *lastName = [record objectForKey:LastNameKey];
    return [lastName length] > 0;
}
```

# Record Filtering

## Set up filters before starting session

# AppControllerSyncing.m

## - sync:

```
- (IBAction)sync:(id)sender
{
      ...
      if (m_syncsUsingRecordFiltering) {
         id filter = [LastNameFilter filter];
         [syncClient setFilters:[NSArray arrayWithObject:filter]];
      }
      else {
         [syncClient setFilters:[NSArray array]];
      }
      ...
}
```

# Making Sync Work

Sync Client

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Record Formatting

## Application stores records differently

# AppControllerSyncing.m

## - convertSyncRecordToAppRecord:withIdentifier:

```
- (void)pullDataForSession:(ISyncSession *)syncSession
{
   ...
   // First phase of two-phase commit.
   NSString *identifier;
   NSEnumerator *enumerator = [pulledIdentifiers objectEnumerator];
   while ((identifier = [enumerator nextObject])) {
      [syncSession clientAcceptedChangesForRecordWithIdentifier:identifier
         formattedRecord:[self myFormatRecord] newRecordIdentifier:nil];
   }
   // Second phase of two-phase commit.
   [syncSession clientCommittedAcceptedChanges];
   [syncSession finishSyncing];
   ...
}
```

# AppControllerSyncing.m

## - convertSyncRecordToAppRecord:withIdentifier:

```
#define FormatLimit 16

- (NSDictionary *)convertSyncRecordToAppRecord:(NSDictionary *)record
     withIdentifier:(NSString *)identifier
{
    ...
    NSString *firstName = [record objectForKey:FirstNameKey];
    ...
    if (m_syncsUsingRecordFormatting) {
       firstName = [firstName length] > FormatLimit ? [firstName
          substringToIndex:FormatLimit] : firstName;
       ...
    }
    ...
}
```

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Sync Alerts

## Sync starting elsewhere… care to join?

# AppController.m

## - awakeFromNib

```
- (void)awakeFromNib
{
    ...
    ISyncClient *client = [self performRegistrations];
    [client setSyncAlertHandler:self selector:
        @selector(client:willSyncEntityNames:)];
    ...
}

// In AppControllerSyncing.m
- (void)client:(ISyncClient *)syncClient
    willSyncEntityNames:(NSArray *)entityNames
{
    [self sync:self];
}
```

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

Feature Decisions

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

*Feature Decisions*

# Demo

## Features

http://developer.apple.com/wwdc2005/

# Making Sync Work

Record filtering?

Record formatting?

Respond to sync alerts?

*Feature Decisions*

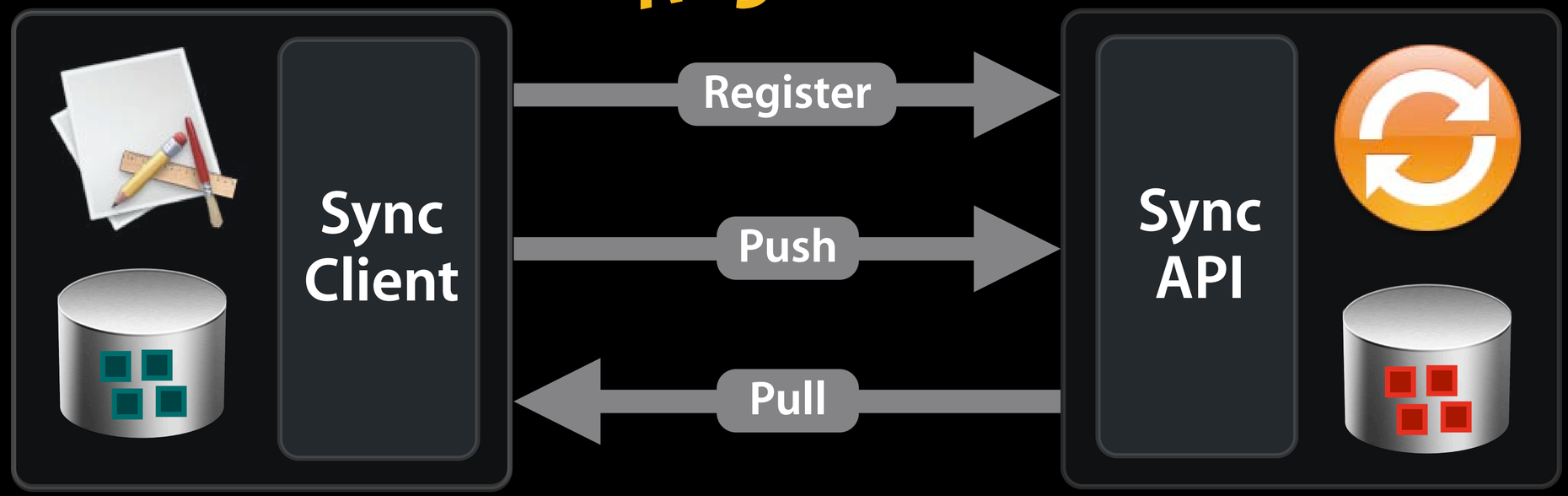
# Making Sync Work

Data Handling

Application

Registration

Sync Engine




Feature Decisions

Session Control Flow

# Whew!

# There’s one more thing…

# Syncrospector

## Developer Tool

# Demo

## Syncrospector

http://developer.apple.com/wwdc2005/

# Syncrospector

## Developer Tool

# Thank you

# More Information

http://developer.apple.com/wwdc2005/

# Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

# Related Sessions

| | |
|---|---|
| Lab: Sync Services | Thursday 2:00PM |

# Q&A

**The Sync Services Team**

http://developer.apple.com/wwdc2005/

# WWDC2005

Worldwide Developers Conference